

In the Claims:

Please cancel claims 1-13, 15-26 and 37, without prejudice, and amend claim 43 as follows:

1-13. Cancelled.

14. Cancelled.

15-26. Cancelled.

27. (Previously presented) A method of compressing a program to be executed by a processor in which compressed-form instructions stored in a program memory are decompressed and cached in an instruction cache prior to being issued, the method comprising:

converting a sequence of original instructions of the program into a corresponding sequence of such compressed-form instructions;

assigning such original instructions imaginary addresses according to said sequence thereof, the assigned imaginary addresses being imaginary addresses at which the instructions are to be considered to exist when held in decompressed form in said instruction cache of the processor; and

outputting a compressed program storable in said program memory and comprising the compressed-form instructions together with imaginary address information specifying said assigned imaginary address of at least one said original instruction so that, when the compressed-form instructions are decompressed and loaded by the processor into the instruction cache, the processor can allocate the assigned imaginary addresses to the decompressed instructions based on said imaginary address information.

28. (Original) A method as claimed in claim 27, wherein the assigned imaginary addresses are selected so that instructions likely to coexist in the instruction cache at execution time will not be mapped to the same cache block.

29. (Previously presented) A method as claimed in claim 27, wherein the compressed-form instructions are arranged to be stored in said program memory in one or more compressed sections, the compressed-form instructions belonging to each section occupying one cache block of the processor's instruction cache when decompressed, and at least one compressed section also containing imaginary address information relating to the instructions of that section.

30. (Previously presented) A method as claimed in claim 29, wherein said

imaginary address information specifies the imaginary address at which a first one of the decompressed instructions corresponding to said at least one compressed section is to be considered to exist when the decompressed instructions are held in said instruction cache.

31. (Original) A method as claimed in claim 29, wherein said imaginary address information is contained in only a first one of said compressed sections to be loaded.

32. (Original) A method as claimed in claim 29, wherein each said compressed section contains imaginary address information relating to the instructions belonging to the section concerned.

33. (Original) A method as claimed in claim 29, wherein the or each said compressed section further contains a decompression key for use by the processor to carry out the decompression of the instructions belonging to said section.

34. (Original) A method as claimed in claim 33, wherein said sequence of original instructions of the program comprises preselected instructions that are not stored explicitly in any said compressed section, and the decompression key of the or each said compressed section identifies the positions at which said preselected instructions exist are to appear in a decompressed sequence of instructions corresponding to the section.

35. (Original) A method as claimed in claim 34, wherein said preselected instructions are “no operation” instructions.

36. (Previously presented) A computer-readable recording medium storing a computer program which carries out a method of compressing a processor program to be executed by a processor, the processor being operable to decompress compressed-form instructions stored in a program memory and to cache the decompressed instructions in an instruction cache prior to issuing them, the computer program comprising:

a converting portion which converts a sequence of original instructions of the processor program into a corresponding sequence of such compressed-form instructions;

an assigning portion which assigns such original instructions imaginary addresses according to said sequence thereof, the assigned imaginary addresses being imaginary address at which the instructions are to be considered to exist when held in decompressed form in said instruction cache of the processor; and

an outputting portion which outputs a compressed program storable in said program memory and comprising the compressed-form instructions together with imaginary address information specifying said assigned imaginary address of at least one said original instruction so that, when the compressed-form instructions are decompressed and loaded by the processor into the instruction cache, the processor can allocate the assigned

imaginary addresses to the decompressed instructions based on said imaginary address information.

37. Cancelled.

38. (Previously added) A processor, for executing instructions of a program stored in compressed form in a program memory, each said compressed-form instruction having an imaginary address at which the instruction is considered to exist when held in decompressed form within the processor, and the program memory also storing imaginary address information from which the imaginary addresses assigned to the compressed-form instructions is derivable, said processor comprising:

a program counter which identifies a position in said program memory;

an instruction cache, having a plurality of cache blocks, each for storing one or more instructions of said program in decompressed form;

an imaginary address deriving unit operable to read the imaginary address information stored in the program memory and to derive therefrom the imaginary address of at least a first one of the compressed-form instructions in said program;

a cache loading unit, comprising a decompression section, operable to perform a cache loading operation in which one or more compressed-form instructions are read from said position in the program memory identified by the program counter and are

decompressed and stored in one of said cache blocks of the instruction cache, which cache block is determined by the imaginary addresses of said one or more compressed-form instructions being read from said position in the program memory;

a cache pointer which identifies a position in said instruction cache of an instruction to be fetched for execution;

an instruction fetching unit which fetches an instruction to be executed from the position identified by the cache pointer and which, when a cache miss occurs because the instruction to be fetched is not present in the instruction cache, causes the cache loading unit to perform said cache loading operation; and

an updating unit which updates the program counter and cache pointer in response to the fetching of instructions so as to ensure that said position identified by said program counter is maintained consistently at the position in said program memory at which the instruction to be fetched from the instruction cache is stored in compressed form.

39. (Previously added) A processor as claimed in claim 38, wherein:

the compressed-form instructions are stored in the program memory in one or more compressed sections, the compressed-form instructions belonging to each section occupying one of said cache blocks when decompressed, and at least one section also contains imaginary address information relating to the instructions belonging to the section; and

said cache loading unit is operable, in said cache loading operation, to decompress and load into one of said cache blocks one such compressed section stored at the position in the program memory identified by the program counter.

40. (Previously added) A processor as claimed in claim 39, wherein said imaginary address information of said at least one section specifies the imaginary address at which a first one of the decompressed instructions corresponding to the compressed section is considered to exist when the decompressed instructions are held in one of the cache blocks.

41. (Previously added) A processor as claimed in claim 39, wherein said imaginary address information is contained in only a first one of said compressed sections to be loaded.

42. (Previously added) A processor as claimed in claim 39, wherein each said compressed section contains imaginary address information relating to the instructions belonging to the section concerned.

43. (Currently amended) A computer-readable recording medium storing a compressed program, said compressed program being adapted to be stored in a program memory of a processor and comprising:

a sequence of compressed-form instructions derived from a corresponding sequence of original instructions, the compressed-form instructions being adapted to be decompressed by the processor and cached in an instruction cache thereof prior to issuance; and

imaginary address information specifying an imaginary address assigned to at least one of said original instructions, being an imaginary address at which that original instruction is to be considered to exist when held in decompressed form in said instruction cache, whereby when the compressed-form instructions are decompressed and loaded by the processor into the instruction cache the processor can allocate the decompressed instructions to such imaginary addresses based on said imaginary address information.

44. (Previously added) A computer-readable recording medium as claimed in claim 43, wherein the assigned imaginary addresses are selected so that instructions likely to coexist in the instruction cache at execution time will not be mapped to the same cache block.

45. (Previously added) A computer-readable recording medium as claimed in claim 43, wherein the compressed-form instructions are arranged to be stored in the said program memory in one or more compressed sections, the compressed-form instructions belonging to each section occupying one cache block of the processor's instruction cache



when decompressed, and at least one compressed section also containing imaginary address information relating to the instructions of that section.

46. (Previously added) A computer-readable recording medium as claimed in claim 45, wherein said imaginary address information specifies the imaginary address at which a first one of the decompressed instructions corresponding to said one compressed section is to be considered to exist when the decompressed instructions are held in the same instruction cache.

47. (Previously added) A computer-readable recording medium as claimed in claim 45, wherein said imaginary address information is contained in only a first one of the said compressed sections to be loaded.

48. (Previously added) A computer-readable recording medium as claimed in claim 45, wherein each said compressed section contains imaginary address information relating to the instructions belonging to the section concerned.

49. (Previously added) A computer-readable recording medium as claimed in claim 45, wherein the or each said compressed section further contains a decompression

key for use by the processor to carry out the decompression of the instructions belonging to the said section.

50. (Previously added) A computer-readable recording medium as claimed in claim 49, wherein said corresponding sequence of original instructions includes preselected instructions that are not stored explicitly in any said compressed section, and the decompression key of the or each said compressed section identifies the positions at which said preselected instructions exist are to appear in a decompressed sequence of instructions corresponding to the section.

51. (Previously added) A computer-readable recording medium as claimed in claim 50, wherein said preselected instructions are “no operation” instructions.